

Federated Learning & Data Privacy, 2024-2025

First Lab – 21 January 2025

Welcome to the first lab session of the Federated Learning & Data Privacy course!

In this lab, we will implement the FedAvg algorithm, understand its components, and explore the impacts of its parameters on the model's performance.

EXERCISE 1 - Get Familiar with the Code

Objective: Understand the structure and components of FedAvg through UML diagramming.

1. **Setup:** Clone the repository TP1 from <https://gitlab.inria.fr/arodio/fedcourse24>.
2. **UML Diagram:** Draw the UML diagram of the provided codebase. Focus on identifying the classes, their attributes, methods, and the relationships between them.
 - Use online tools like ChatGPT to generate the diagram.
 - Use online tools like www.planttext.com for creating the UML diagram.
 - Add brief descriptions to each component to explain its functionality.

Attach here the UML Diagram

EXERCISE 2 – Complete the code

Objective: Understand the core mechanics of FedAvg by implementing the missing code.

1. **Client Local Training:**
 - Complete the `step` method in `client.py` to perform local training on client data.
 - Complete the `write_logs` method in `client.py` to record training metrics.
2. **Aggregator Orchestration:**

- Complete the `mix` method in `aggregator.py` to aggregate client models.
- Complete the `update_clients` method in `aggregator.py` to synchronize the updated global model with the clients.

3. Training Process:

- Implement the training loop in `train.py` to run the FedAvg algorithm.

4. Run the Experiment:

- Once you've completed the code, run the FedAvg algorithm using the script `run.sh` and observe the output.

Attach your code here

EXERCISE 3 - The Effect of Local Epochs

Objective: Analyze how the number of local epochs affects the model's performance in a federated learning setting.

1. Experiment:

- Run FedAvg for different numbers of local epochs (e.g., 1, 5, 10, 50, 100).
- Record the test accuracy for each setting.

2. Plot:

- Create a plot with the local epochs on the x-axis and test accuracy on the y-axis.
- Interpret the plot and draw your conclusions. How does the number of local epochs influence the learning process and the final model accuracy? Were you expecting this result? Motivate your answer.

Attach your plot here.

BONUS EXERCISE - Complex Dataset & Model

Objective: Challenge yourself by integrating a more complex dataset and a deep learning model into the framework.

1. Implementation:

- Integrate the CIFAR10 dataset into the learning process.
- Implement the MobileNet model using the PyTorch framework.

2. Training & Evaluation:

- Train the model using the FedAvg algorithm and evaluate its performance.

Good luck, and don't hesitate to ask questions and collaborate with your peers!

At the end of the lesson, you can send your document to: francesco.diana@inria.fr